1. 题目

LuoguP1255 数楼梯

dp, bfs, <https://www.luogu.com.cn/problem/P1255>

代码：

n = int(input())

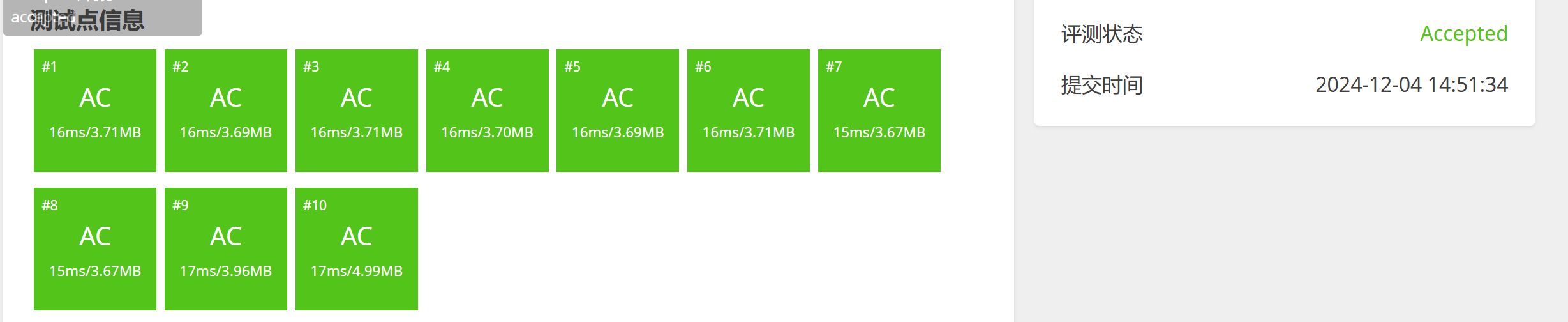
dp = [1] \* (n + 1)

for i in range(2, n + 1):

dp[i]=dp[i - 1]+dp[i - 2]

print(dp[n])

代码运行截图 （至少包含有"Accepted"）



27528: 跳台阶

dp, <http://cs101.openjudge.cn/practice/27528/>

代码：

n = int(input())

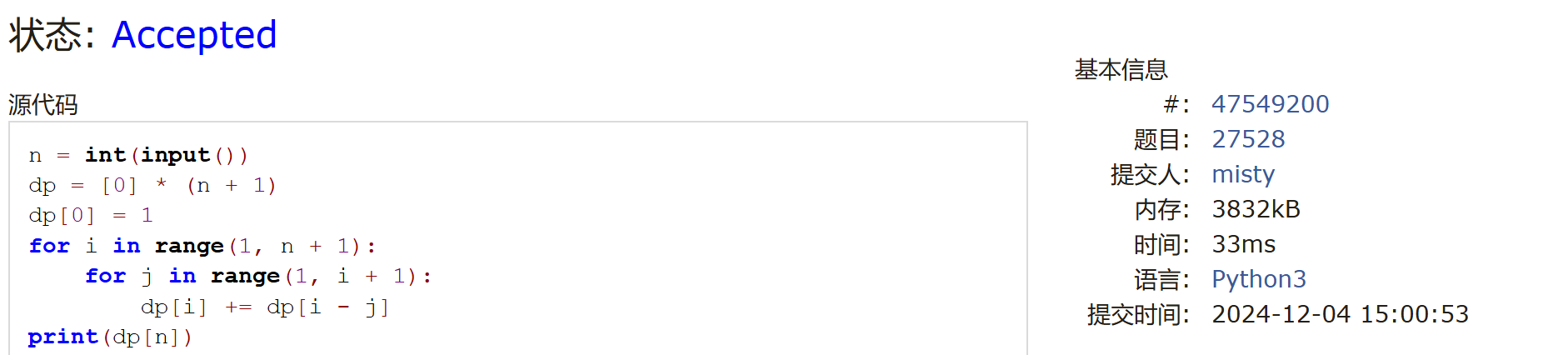
dp = [0] \* (n + 1)

dp[0] = 1for i in range(1, n + 1):

for j in range(1, i + 1):

dp[i] += dp[i - j]print(dp[n])

代码运行截图 ==（至少包含有"Accepted"）==



474D. Flowers

dp, <https://codeforces.com/problemset/problem/474/D>

代码：

P = int(1e9) + 7

def main():

n = int(1e5)

T, k = map(int, input().split())

f = [1] \* (n + 1)

s = [i + 1 for i in range(n + 1)]

for i in range(k, n + 1):

f[i] = (1 + s[i - k]) % P

s[i] = (s[i - 1] + f[i]) % P

for \_\_\_ in range(T):

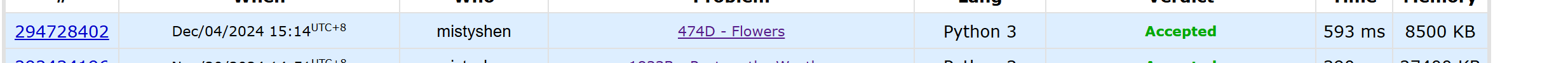
x, y = map(int, input().split())

print(((s[y] - s[x - 1]) % P + P) % P)

if \_\_name\_\_ == "\_\_main\_\_":

main()

代码运行截图 （至少包含有"Accepted"）



LeetCode5.最长回文子串

dp, two pointers, string, <https://leetcode.cn/problems/longest-palindromic-substring/>

代码：

class Solution:

    def expandAroundCenter(self, s, left, right):

        while left >= 0 and right < len(s) and s[left] == s[right]:

            left -= 1

            right += 1

        return left + 1, right - 1

    def longestPalindrome(self, s: str) -> str:

        start, end = 0, 0

        for i in range(len(s)):

            left1, right1 = self.expandAroundCenter(s, i, i)

            left2, right2 = self.expandAroundCenter(s, i, i + 1)

            if right1 - left1 > end - start:

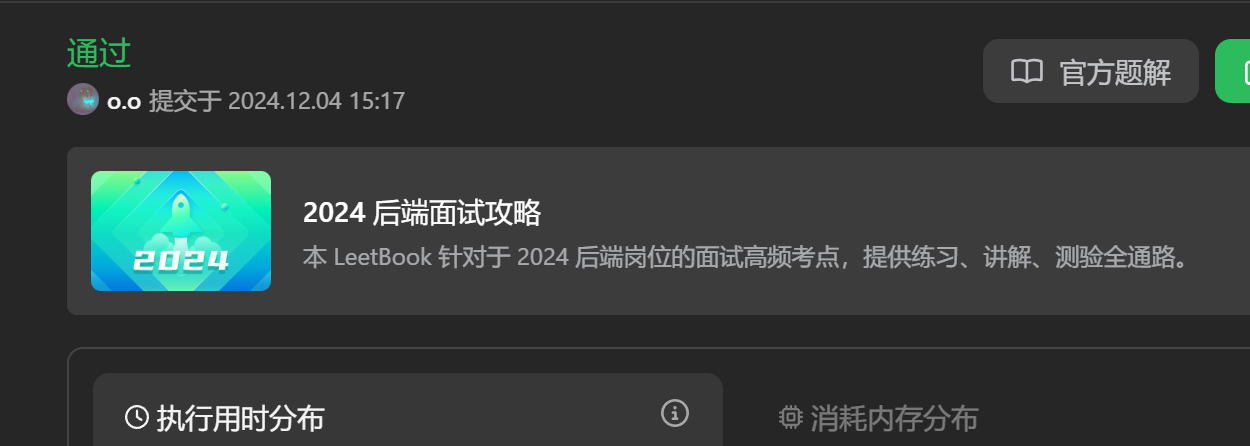
                start, end = left1, right1

            if right2 - left2 > end - start:

                start, end = left2, right2

        return s[start: end + 1]

代码运行截图 （至少包含有"Accepted"）



12029: 水淹七军

bfs, dfs, <http://cs101.openjudge.cn/practice/12029/>

代码：

import sys

sys.setrecursionlimit(300000)input = sys.stdin.read

# 判断坐标是否有效def is\_valid(x, y, m, n):

return 0 <= x < m and 0 <= y < n

# 深度优先搜索模拟水流def dfs(x, y, water\_height\_value, m, n, h, water\_height):

dx = [-1, 1, 0, 0]

dy = [0, 0, -1, 1]

for i in range(4):

nx, ny = x + dx[i], y + dy[i]

if is\_valid(nx, ny, m, n) and h[nx][ny] < water\_height\_value:

if water\_height[nx][ny] < water\_height\_value:

water\_height[x][y] = water\_height\_value

dfs(nx, ny, water\_height\_value, m, n, h, water\_height)

# 主函数def main():

data = input().split() # 快速读取所有输入数据

idx = 0

k = int(data[idx])

idx += 1

results = []

for \_ in range(k):

m, n = map(int, data[idx:idx + 2])

idx += 2

h = []

for i in range(m):

h.append(list(map(int, data[idx:idx + n])))

idx += n

water\_height = [[0] \* n for \_ in range(m)]

i, j = map(int, data[idx:idx + 2])

idx += 2

i, j = i - 1, j - 1

p = int(data[idx])

idx += 1

for \_ in range(p):

x, y = map(int, data[idx:idx + 2])

idx += 2

x, y = x - 1, y - 1

if h[x][y] <= h[i][j]:

continue

dfs(x, y, h[x][y], m, n, h, water\_height)

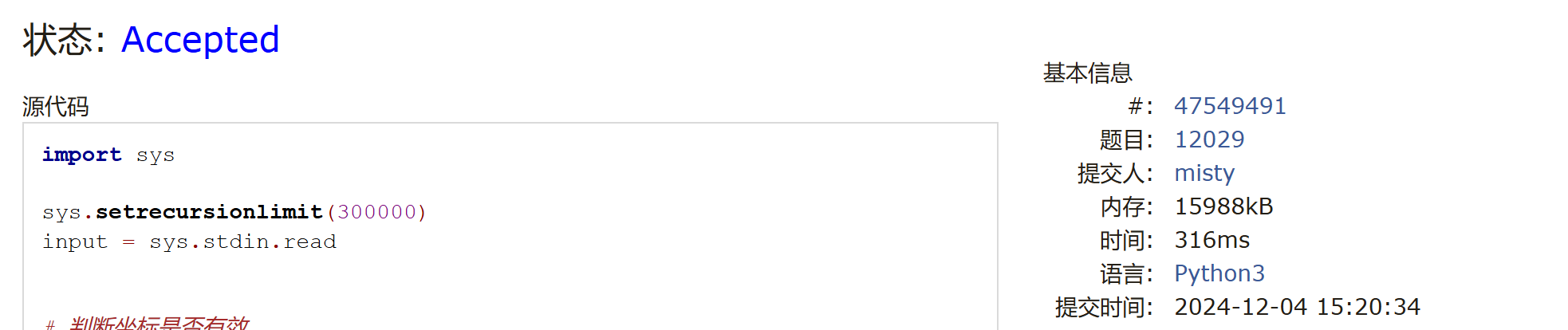
results.append("Yes" if water\_height[i][j] > 0 else "No")

sys.stdout.write("\n".join(results) + "\n")

if \_\_name\_\_ == "\_\_main\_\_":

main()

代码运行截图 （至少包含有"Accepted"）



02802: 小游戏

bfs, <http://cs101.openjudge.cn/practice/02802/>

代码：

from collections import dequefrom collections import defaultdict

def bfs(start, end, grid, h, w):

queue = deque([start])

in\_queue = defaultdict(lambda: float('inf'))

dirs = [(0, -1), (-1, 0), (0, 1), (1, 0)]

min\_x = float('inf')

while queue:

x, y, d, seg = queue.popleft()

for i, (dx, dy) in enumerate(dirs):

nx, ny = x + dx, y + dy

new\_seg = seg if i == d else seg + 1

if (nx, ny) == end:

min\_x = min(min\_x, new\_seg)

continue

if (0 <= nx < h + 2 and 0 <= ny < w + 2 and new\_seg<in\_queue[(nx,ny,i)]

and grid[nx][ny] != 'X'):

in\_queue[(nx, ny, i)] = new\_seg

queue.append((nx, ny, i, new\_seg))

return min\_x

board\_num = 1while True:

w, h = map(int, input().split())

if w == h == 0:

break

grid = [' ' \* (w + 2)] + [' ' + input() + ' ' for \_ in range(h)] + [' ' \* (w + 2)]

print(f"Board #{board\_num}:")

pair\_num = 1

while True:

y1, x1, y2, x2 = map(int, input().split())

if x1 == y1 == x2 == y2 == 0:

break

start = (x1, y1, -1, 0)

end = (x2, y2)

seg = bfs(start, end, grid, h, w)

if seg == float('inf'):

print(f"Pair {pair\_num}: impossible.")

else:

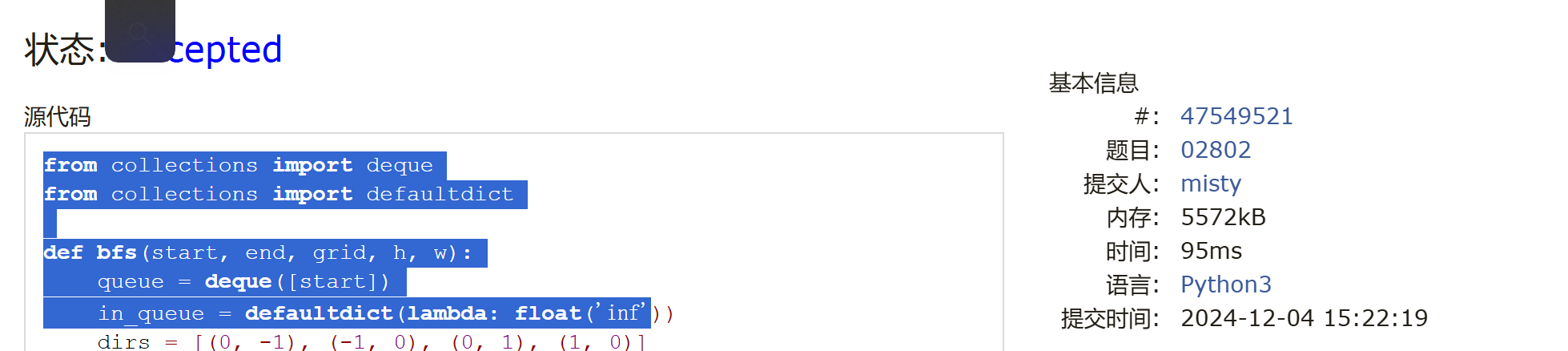
print(f"Pair {pair\_num}: {seg} segments.")

pair\_num += 1

print()

board\_num += 1

代码运行截图 （至少包含有"Accepted"）



2. 学习总结和收获

1.DP会做了

2.bfs还不会做

3.下周上机去熟悉一下机房环境